AMENDMENTS TO THE SPECIFICATION

Please replace paragraph [0009] with the following amended paragraph:

]0009] Several classes of algorithms are currently used to encrypt and decrypt data. Algorithms according to one such class (i.e., public key cryptographic algorithms, an instance of which is the Rivest, Shamir, Adlemen (RSA) algorithm~~RSA algorithm~~) employ two cryptographic keys, a public key and a private key, to encrypt or decrypt data. According to some of the public key algorithms, a recipient's public key is employed by a sender to encrypt data for transmission to the recipient. Because there is a mathematical relationship between a user's public and private keys, the recipient must employ his private key to decrypt the transmission in order to recover the data. Although this class of cryptographic algorithms enjoys widespread use today, encryption and decryption operations are exceedingly slow even on small amounts of data. A second class of algorithms, known as symmetric key algorithms, provide commensurate levels of data security and can be executed much faster. These algorithms are called symmetric key algorithms because they use a single cryptographic key to both encrypt and decrypt information. In the public sector, there are currently three prevailing single-key cryptographic algorithms: the Data Encryption Standard (DES), Triple DES, and the Advanced Encryption Standard (AES). Because of the strength of these algorithms to protect sensitive data, they are used now by U.S. Government agencies, but it is anticipated by those in the art that one or more of these algorithms will become the standard for commercial and private transactions in the near future. According to all of these symmetric key algorithms, plaintext and ciphertext is divided into blocks of a specified size for encryption and decryption. For example, AES performs cryptographic operations on blocks 128 bits in size, and uses cryptographic key sizes of 128-, 192-, and 256-bits. Other symmetric key algorithms such as the Rijndael Cipher allow for 192- and 256-bit data blocks as well. Accordingly, for a block encryption operation, a 1024-bit plaintext message is encrypted as eight 128-bit blocks.

Please replace paragraphs [0012] through [0014] with the following amended paragraphs:

]0012] To perform cryptographic operations on multiple successive blocks of text, all of the symmetric key algorithms employ the same types of modes. These modes include electronic code book (ECB) mode, cipher block chaining (CBC) mode, cipher feedback (CFB) mode, and output feedback (OFB) mode. Some of these modes utilize an additional initialization vector during performance of the sub-operations and some use the ciphertext output of a first set of cryptographic rounds performed on a first block of plaintext as an additional input to a second set of cryptographic rounds performed on a second block of plaintext. It is beyond the scope of the present application to provide an in depth discussion of each of the cryptographic algorithms and sub-operations employed by present day symmetric key cryptographic algorithms. For specific implementation standards, the reader is directed to *Federal Information Processing Standards Publication 46-3* (FIPS-46-3), dated October 25, 1999 for a detailed discussion of DES and Triple DES, and *Federal Information Processing Standards Publication 197* (FIPS-197), dated November 26, 2001 for a detailed discussion of AES. Both of the aforementioned standards are issued and maintained by the National Institute of Standards and Technology (NIST) and are herein incorporated by reference for all intents and purposes. In addition to the aforementioned standards, tutorials, white papers, toolkits, and resource articles can be obtained from NIST's Computer Security Resource Center (CSRC) over the Internet ~~at http://csrc.nist.gov/~~.

]0013] One skilled in the art will appreciate that there are numerous application programs available for execution on a computer system that can perform cryptographic operations (i.e., encryption and decryption). In fact, some operating systems (e.g. Microsoft® WindowsXP®, LINUX®~~Linux~~) provide direct encryption/decryption services in the form of cryptographic primitives, cryptographic application program interfaces, and the like. The present inventors, however, have observed that present day computer cryptography techniques are deficient in several respects. Thus, the reader's attention is directed to FIGURE 1, whereby these deficiencies are highlighted and discussed below.

]0014] FIGURE 1 is a block diagram 100 illustrating present day computer cryptography applications. The block diagram 100 depicts a first computer workstation 101 connected to a local area network 105. Also connected to the network 105 is a second computer

workstation 102, a network file storage device 106, a first router 107 or other form of interface to a wide area network (WAN) 110 such as the Internet, and a wireless network router 108 such as one of those compliant with Institute of Electrical and Electronics Engineers (IEEE) Standard~~IEEE Standard~~ 802.11. A laptop computer 104 interfaces to the wireless router 108 over a wireless network 109. At another point on the wide area network 110, a second router 111 provides interface for a third computer workstation 103.

Kindly replace paragraph [0019] with the following amended paragraph:

[0019] Furthermore, the present inventors have noted that the accomplishment of cryptographic operations on a present day computer system 101-104 is very much analogous to the accomplishment of floating point mathematical operations prior to the advent of dedicated floating point units within microprocessors. Early floating point operations were performed via software and hence, they executed very slowly. Like floating point operations, cryptographic operations performed via software are disagreeably slow. As floating point technology evolved further, floating point instructions were provided for execution on floating point co-processors. These floating point co-processors executed floating point operations much faster than software implementations, yet they added cost to a system. Likewise, cryptographic co-processors exist today in the form of add-on boards or external devices that interface to a host processor via parallel ports or other interface buses (e.g., Universal Serial Bus (USB))~~(e.g., USB)~~. These co-processors certainly enable the accomplishment of cryptographic operations much faster than pure software implementations. But cryptographic co-processors add cost to a system configuration, require extra power, and decrease the overall reliability of a system. Cryptographic co-processor implementations are additionally vulnerable to snooping because the data channel is not on the same die as the host microprocessor.

Please delete the section entitled "SUMMARY OF THE INVENTION" in its entirety and substitute the following section therefor:

SUMMARY OF THE INVENTION

~~The present invention,~~[0020.1] The present invention, among other applications, is directed to solving these and other problems and disadvantages of the prior art. The present invention provides a superior technique for performing cryptographic operations within a microprocessor. In one embodiment, an apparatus for performing cryptographic operations is provided. The apparatus includes a cryptographic instruction ~~and execution~~and execution logic. The cryptographic instruction is received by a ~~computing device~~microprocessor as part of an instruction flow executing on the microprocessor~~computing device~~. The cryptographic instruction prescribes one of the cryptographic operations, and also prescribes that an intermediate result be generated. The execution logic is operatively coupled to the cryptographic instruction. The execution logic executes the one of the cryptographic operations, and generates the intermediate result.

[0021] One aspect of the present invention contemplates an apparatus for performing cryptographic operations. The apparatus has a control word and a cryptography unit. The control word prescribes that an intermediate result be generated during execution of one of the cryptographic operations. The cryptography unit is within a microprocessor ~~device~~ and is configured to execute the one of the cryptographic operations responsive to receipt of a cryptographic instruction within an instruction flow that prescribes the one of the cryptographic operations, where the cryptographic instruction also references the control word.

[0022] Another aspect of the present invention provides a method for performing cryptographic operations ~~in a device~~. The method includes, via a cryptographic instruction, prescribing that an intermediate result be generated during execution of one of a plurality of cryptographic operations; and, within a microprocessor, receiving the cryptographic instruction, and generating the intermediate result when executing the one of the cryptographic operations.